Sisteme de Prelucrare Grafică

Curs nr. 2

1.2. Transformări de coordonate

Considerăm două sisteme de coordonate în plan xOy şi x'O'y'. Unuipunct fix din plan, P, îi corespund două reprezentări: (x,y) în sistemul xOy şi (x',y') în sistemul x'O'y'

a) Translaţia

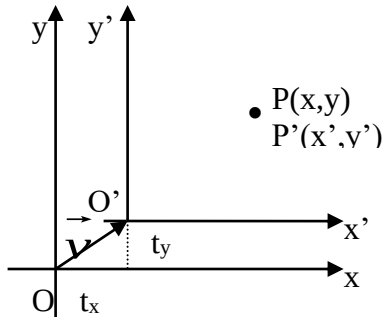


Fig.1

Dacă sistemul x'O'y' s-a abţinut prin translaţia sistemului xOy cu vectorul de translaţie (Fig.1)

$$\vec{v} = t_x * \vec{i} + t_y * \vec{j}$$

atunci relaţia dintre coordonatele lui P în cele două sisteme este:

$$\begin{cases} x' = x - t_x \\ y' = y - t_y \end{cases} \qquad (1)$$

$$T_C(t_x, t_y) = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad T(-t_x,-t_y) = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow T_C(t_x, t_y) = [T(t_x, t_y)]^{-1} = T(-t_x,-t_y) \qquad (2)$$

$T_C(t_x,t_y)$ – matricea transformării de coordonate (în coordonate omogene)
$T(t_x,t_y)$ – matricea transformării geometrice

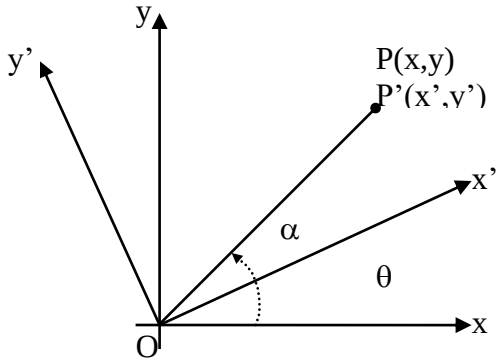
Obs: Matricea transformării de coordonate este inversa matricii transformării geometrice.

b) Rotaţia

Sistemul x'O'y' obţinut prin rotaţia axelo lui xOy cu unghiul θ :

$$P: \begin{cases} x = r\cos\alpha \\ y = r\sin\alpha \end{cases} \quad în \quad xOy$$

$$P': \begin{cases} x' = r\cos(\alpha - \theta) \\ y' = r\sin(\alpha - \theta) \end{cases} \quad în \quad x'O'y'$$



$$\Rightarrow \begin{cases} x' = r\cos\alpha\cos\theta + r\sin\alpha\sin\theta = x\cos\theta + y\sin\theta \\ y' = r\sin\alpha\cos\theta - r\cos\alpha\sin\theta = -x\cos\theta + y\cos\theta \end{cases}$$

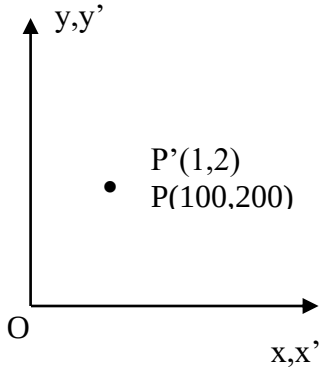
$\Rightarrow R_C(\theta)$-matricea transformării în cazul rotaţiei unui sistem de coordonate

$$R_C(\theta)=[R(\theta)]^{-1}=R(-\theta) \qquad (3)$$

R-rotaţia geometrică

c) Scalarea

Sistemele xOy şi x'O'y' au aceleaşi axe, aceeaşi origine, doar o altă unitate de măsură.



De exemplu:

$$x' = \frac{1}{sX}x, \qquad y' = \frac{1}{sY}y \qquad \Rightarrow$$

$$S_C(s_x,s_y)=[S(s_x,s_y)]^{-1}= S(\frac{1}{s_x},\frac{1}{s_y}) \qquad (4)$$

$$S_C(s_x, s_y) = \begin{bmatrix} \frac{-1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}; \qquad S(s_x, s_y) = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1.3. Transformarea de instanţiere

Obiectele se definesc în mod obişnuit într-un sistem de coordonate propriu (local – sistem de coordonate obiect). Operaţia de trecere din sistemul local în alt spaţiu de desenare (care are altă origine şi alte coordonate), se numeşte instanţiere, iar transformarea respectivă – transformare de instanţiere.

Reprezentarea în diverse poziţii, dimensiuni şi orientări în noul spaţiu se numeşte instanţă a obiectului.

Transformarea de instanţiere este compusă, în general, din: scalare, rotaţii şi translaţie.

Obs: Translaţia este ultima transformare care se aplică.

OpenGL foloseşte aceste transformări în plasarea obiectelor în scene 3D.

1.4. Transformarea de vizualizare 2D

Transformarea dintr-un sistem 2D oarecare, numit sistem al lumii reale WCS – World Coordinate System, în spaţiul de afişare dispozitiv (sistem de coordonate dispozitiv, sistem de coordonate fizice), care este propriu fiecărui dispozitiv, se numeşte transformare de vizualizare.

Obs: Sistemul WCS se mai numeşte SC logice sau SC universale sau SC utilizator.

Cele două sisteme 2D se specifică prin două dreptunghiuri, transformarea de vizualizare având ca efect transformarea fiecărui punct din dreptunghiul 2D din SC logice în dreptunghiul din SC dispozitiv (spaţiul de afişare).

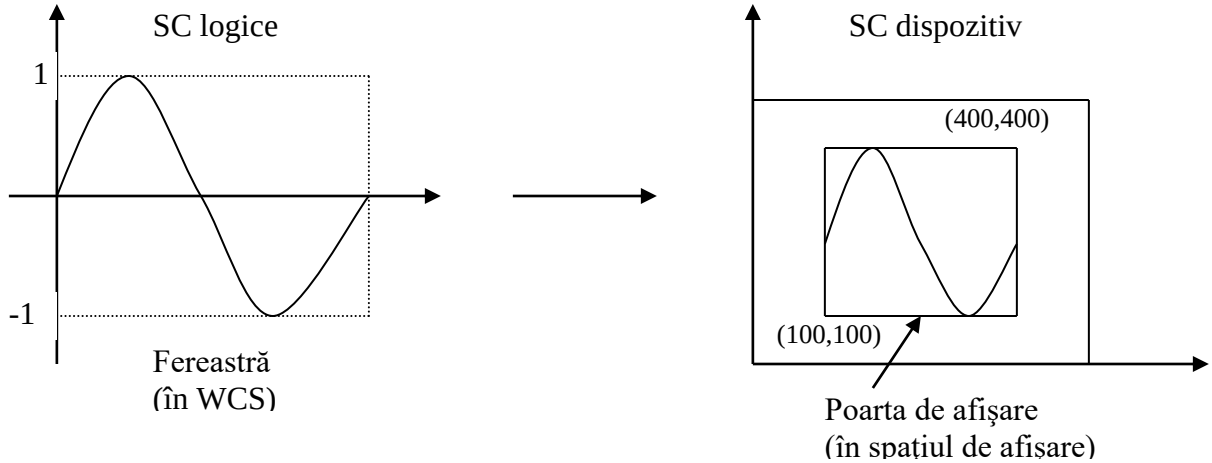


Fig.1

Transformata de vizualizare se mai numeşte şi transformata fereastră-poartă.

Considerăm mai întâi aceeaşi orientare a axelor celor două SC. Fie $F(x_f,y_f)$ un punct din fereastră şi $P(x_p,y_p)$ punctul corespunzător din poartă. Transformarea fereastră-poartă este definită astfel încât poziţia relativă a punctului P în poartă să fie aceeaşi cu poziţia relativă a punctului F în fereastră.

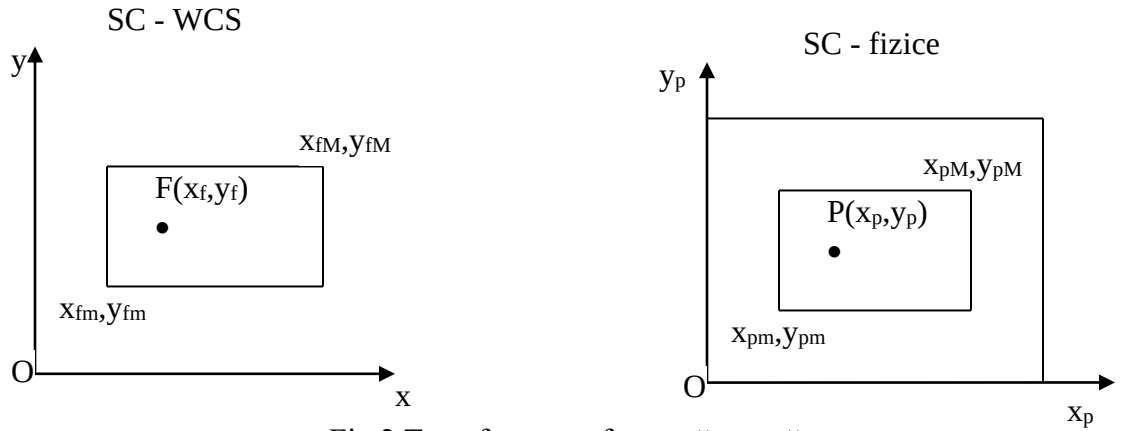


Fig.2 Transformarea fereastră-poartă

$$\frac{x_p - x_{pm}}{x_{pM} - x_{pm}} = \frac{x_f - x_{fm}}{x_{fM} - x_{fm}} \quad , \quad \frac{y_p - y_{pm}}{y_{pM} - y_{pm}} = \frac{y_f - y_{fm}}{y_{fM} - y_{fm}} \qquad (5)$$

Notăm: $$\begin{cases} s_x = \dfrac{\Delta x_p}{\Delta x_f} = \dfrac{x_{pM} - x_{pm}}{x_{fM} - x_f} & (6) \\ s_y = \dfrac{\Delta y_p}{\Delta y_f} = \dfrac{y_{pM} - y_{pm}}{y_{fM} - y_f} & (7) \end{cases}$$

$$\Rightarrow \begin{cases} \dfrac{x_p - x_{pm}}{x_f - x_{fm}} = s_x \\ \dfrac{y_p - y_{pm}}{y_f - y_{fm}} = s_y \end{cases} \Rightarrow \begin{cases} x_p = x_f * s_x + x_{pm} - x_{fm} * s_x \\ y_p = y_f * s_y + y_{pm} - y_{fm} * s_y \end{cases} \qquad (8)$$

Notăm: $$\begin{cases} t_x = x_{pm} - x_{fm} * s_x \\ t_y = y_{pm} - y_{fm} * s_y \end{cases} \qquad (9)$$

$$\Rightarrow \begin{cases} x_p = x_f * s_x + t_x \\ y_p = y_f * s_y + t_y \end{cases} \qquad (10)$$

Formulele (8) şi (10) sunt formule de transformare fereastră-poartă când sistemele de coordonate au aceeaşi orientare a axelor.

În coordonate omogene $\Rightarrow$ $P = T * S * F = Mv * F$

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \qquad (11)$$ Formule de transformare fereastră-poartă

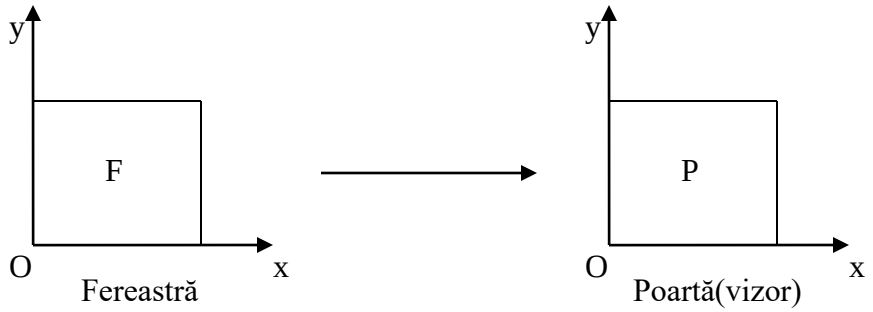
M_V – matricea transformării de vizualizare este:

$$M_V = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (12)$$

Obs: Întâi se realizează scalarea şi apoi translarea $T * S \neq S * T$

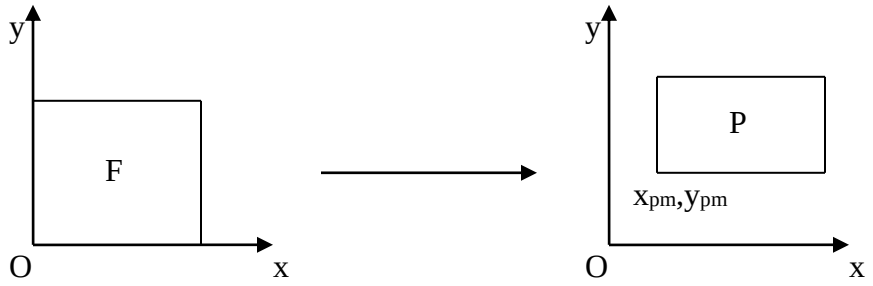
Cazuri particulare

1) $t_x=t_y=0$ (nu există translaţie) $\begin{pmatrix} x_{pm}=0, x_{fm}=0 \\ y_{pm}=0, y_{fm}=0 \end{pmatrix}$

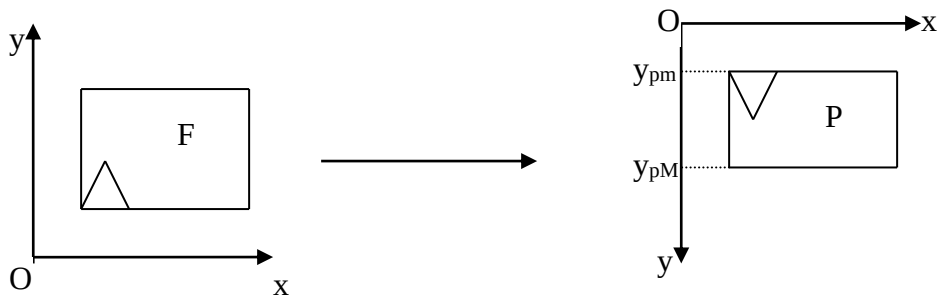


2) $t_x=x_{pm}$ $\begin{pmatrix} x_{fm}=0 \\ y_{fm}=0 \end{pmatrix}$

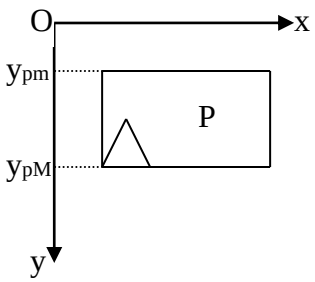
$t_y=y_{pm}$



3) Sistemul de coordonate al ecranului are originea în colţul stânga-sus. Aplicând formulele (10) desenul definit în fereastră va fi răsturnat în poartă.



Trebuie să avem:



$\Rightarrow$că coordonata y_p trebuie să se modifice astfel:

$$\begin{cases} x_p = x_f * s_x + t_x \\ y_p = y_{pM} + y_{pm} - (y_f * s_y + t_y) = y_{pM} + y_{pm} - y_f * s_y - y_{pm} + y_{fm} * s_y \end{cases}$$

În concluzie, pentru ecran vom avea:

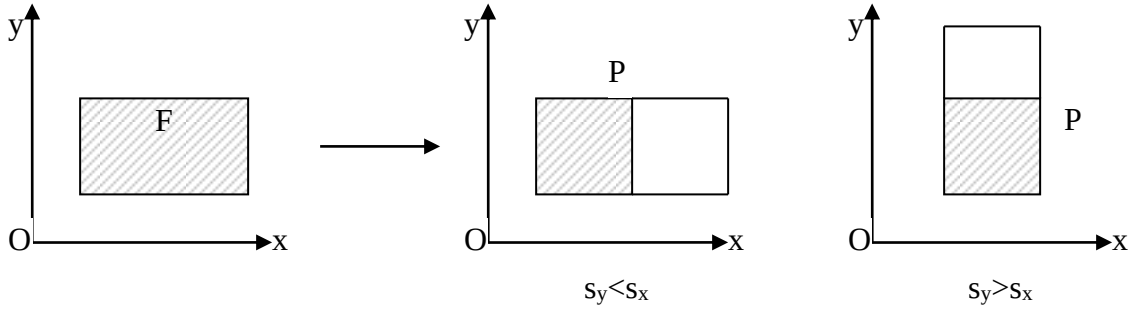
$$\begin{cases} x_p = x_f * s_x + t_x \\ y_p = y_{pM} - y_f * s_y + y_{fm} * s_y \end{cases} \qquad (13)$$

Obs: Când fereastra şi poarta nu sunt asemenea are loc o deformare a obiectelor afişate în poartă. În unele aplicaţii se doreşte deformarea, în altele nu.

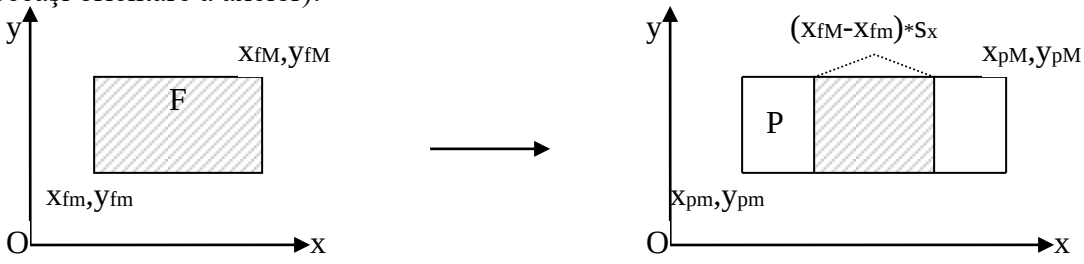
Dacă nu se doreşte deformarea se calculează s_x şi s_y şi se alege valoarea minimă, astfel încât scalarea să fie uniformă.

$$s = \min(s_x, s_y)$$

În acest caz zona definită de fereastră ocupă zona din stânga a porţii sau zona de jos a porţii:

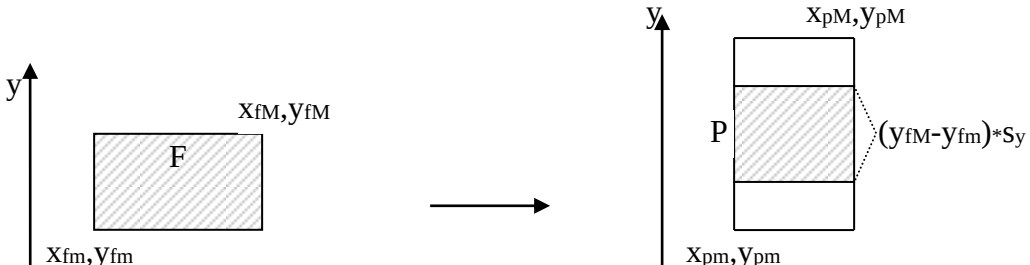


Dacă dorim ca imaginea să apară centrată în poartă putem avea cazurile (considerând aceeaşi orientare a axelor):



Fereastră Poartă

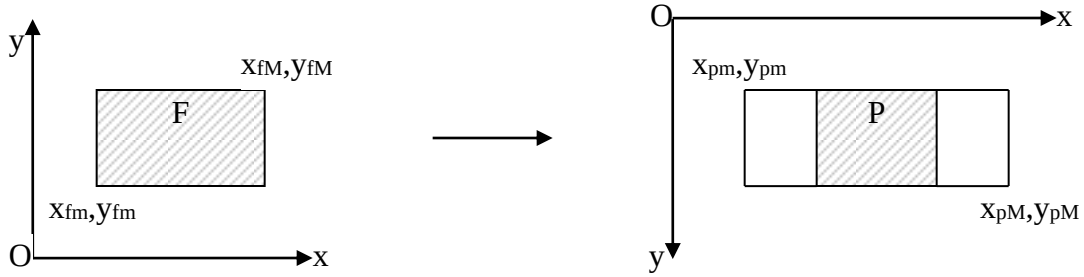
sau



Fereastră Poartă

$$\begin{cases} x_p = x_f * s_x + t_x + \dfrac{x_{pM} - x_{pm} - (x_{fm} - x_{fm}) * s_x}{2} \\ y_p = y_f * s_y + t_y + \dfrac{y_{pM} - y_{pm} - (y_{fm} - y_{fm}) * s_y}{2} \end{cases} \qquad (14)$$

În <u>cazul ecranului, axa y fiind în jos</u> rezultă:



$$\begin{cases} x_p = x_f * s_x + t_x' \\ y_p = y_{pM} + y_{pm} - (y_f * s_y + t_y') \end{cases} \qquad (15)$$

unde $\begin{cases} t_x' = x_{pm} - x_{fm} * s_x + \dfrac{\Delta x_p - s_x * \Delta x_f}{2} \\ t_y' = y_{pm} - y_{fm} * s_y + \dfrac{\Delta y_p - s_y * \Delta y_f}{2} \end{cases}$

Obs:

1) Dacă se foloseşte funcţia ***setviewport(*x_{pmin}*,*y_{pmin}*,*x_{pmax}*,*y_{pmax}*,decup)*** originea sistemului de coordonate al ecranului este considerată a fi punctul ***(*x_{pmin}*,*y_{pmin}*)***. În consecinţă din formulele (15) se va scădea x_{pmin} din prima şi y_{pmin} din a doua.

2) În cazul ecranului se pot folosi formulele (8) şi (14) care sunt determinate pentru aceeaşi orientare a axelor şi apoi se efectuează o rotaţie cu π pentru a nu obţine imaginea răsturnată.